ESnet
ENERGY SCIENCES NETWORK

ESnet6 High Touch Services

Precision Streaming Network Telemetry

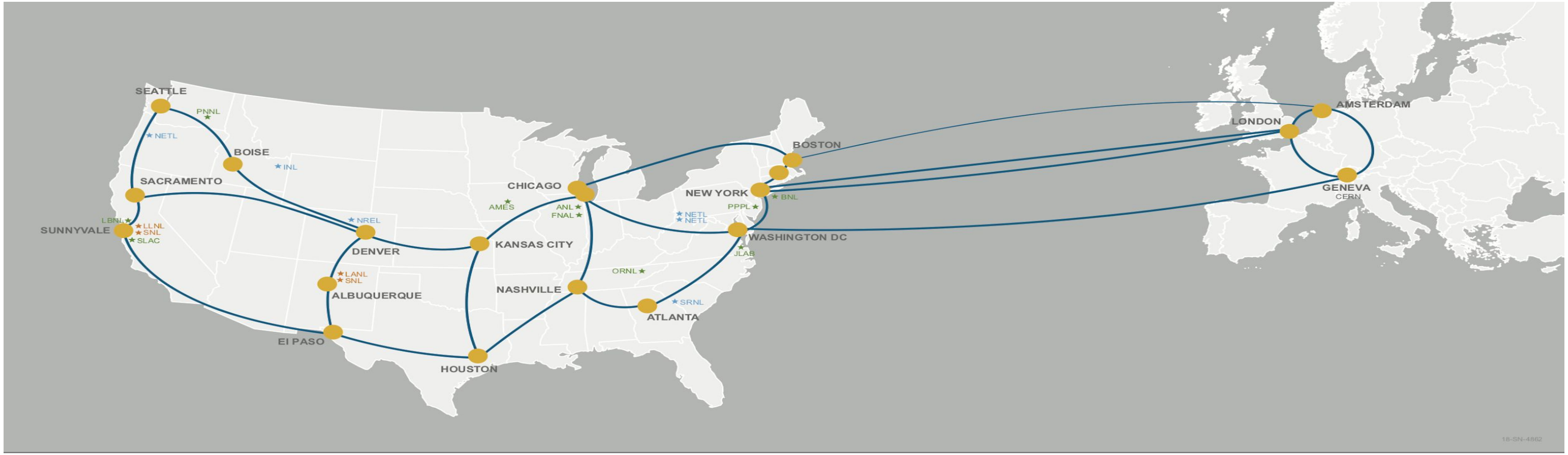
Bruce A. Mah, Richard Cziva, Chin Guok, Yatish Kumar
Energy Sciences Network
Lawrence Berkeley National Laboratory

September 25, 2020

U.S. DEPARTMENT OF ENERGY
Office of Science

BERKELEY LAB

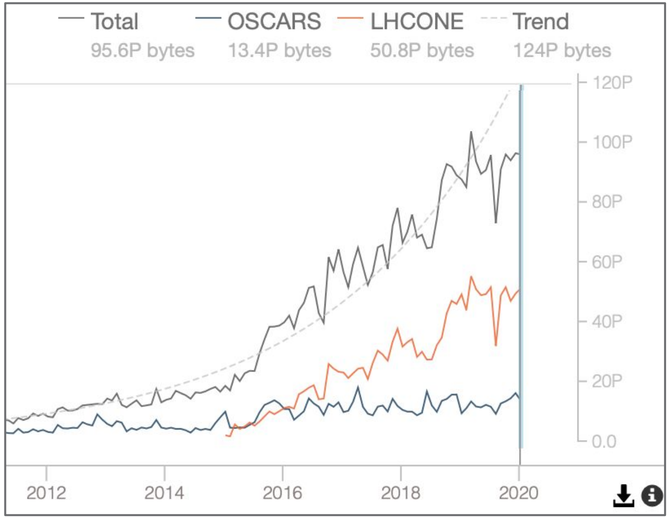
ESnet: DOE’s high-performance network (HPN) user facility optimized for enabling big-data science



ESnet provides connectivity to
all of the DOE labs, experiment sites, & supercomputers

Increased Need for Programmability

- ESnet's traffic, user-base and the experiments continue to grow in a fast pace
- Computing and data model are also evolving, requiring:
  - fine-grained visibility in real-time
  - application-specific traffic handling
  - programmable, in-network services
- Needs not addressed by existing measurement mechanisms (sampled, aggregated, delayed)
- High Touch Services created to fulfill these needs



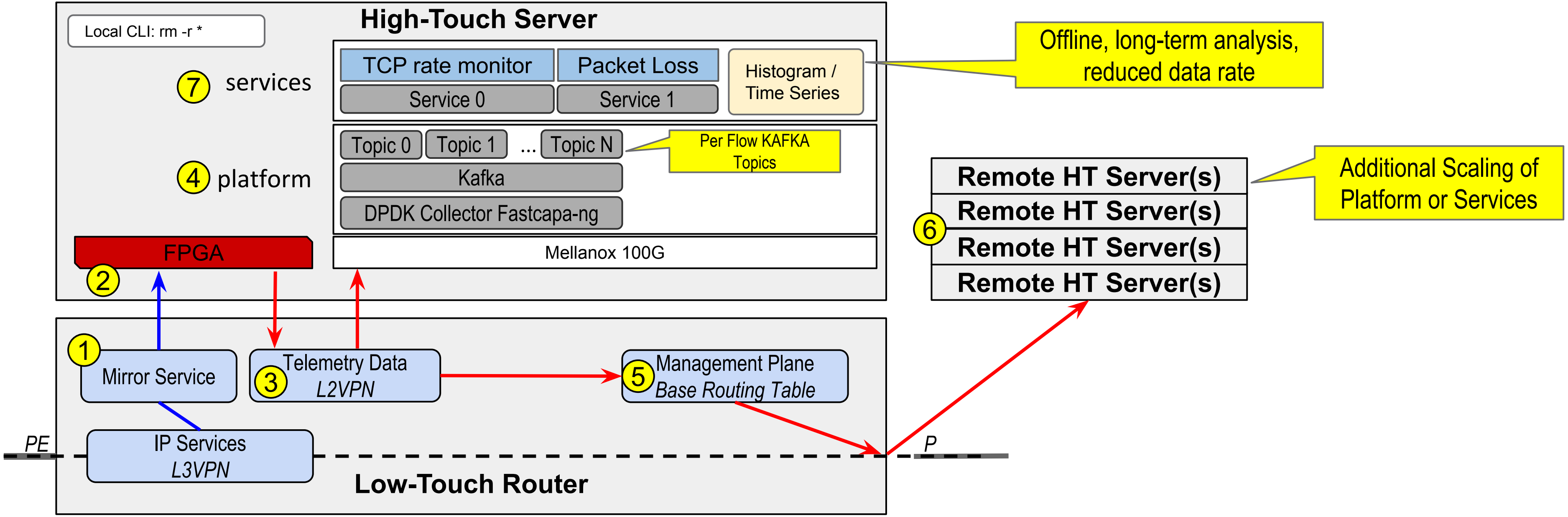
Live ESnet usage statistics: my.es.net
Total carried: Exabyte/year.

ESnet

High-Touch Services

- High-precision, real-time visibility into network traffic
  - Process every packet of interest in real-time
  - Accurate, precision timing (ns precision / accuracy)
  - Software-defined functionality
  - Programmatically deployable and customizable
- In contrast to "low touch" services
  - Fixed function services such as IP packet routing, basic statistics
  - Optimized for speed and low cost, but not flexible
- Technology enablers
  - Software-defined networking
  - Programmable network dataplane hardware with accurate timestamps
  - High-speed packet processing libraries (DPDK, etc.)

ESnet6 High-Touch Architecture Overview



1. Mirror Service - Allows selective flows in the dataplane to be duplicated and sent to the FPGA for processing.
2. Programmable Dataplane (DP) - Appends meta-data, timestamps and repackages packet for transmission to Platform code.
3. Telemetry Data L2VPN - Connect Dataplane and Platform, possibly on different High-Touch Servers.
4. Platform - Reads telemetry packets from the network and distributes information to High Touch Services.
5. Management Plane Base Routing Table - Provides connectivity to Remote Servers.
6. Remote Server - Hosts Platform components or Services (but not a Dataplane). Telemetry data can be directed to Remote Servers.
7. Service - Reads data from the Platform and performs real-time analysis as well as inserts selected telemetry data into database.

Datapath of Customer Packet
Datapath of Mirrored Packet
Datapath of Telemetry Packet

ESnet

What Programmable "High Touch" Hardware to Use?

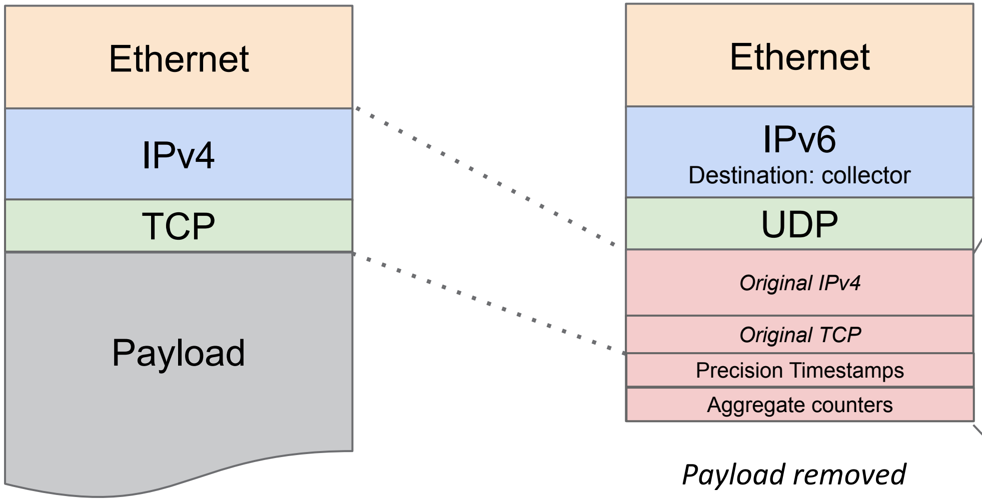
- There are a variety of programmable network devices available today. ESnet was looking for the following:
  - 100Gbit/s port speed and roadmap for higher speeds
  - Timing and performance guarantees
  - Easy programming (P4 style)
  - Established vendor with support
- We are currently prototyping using Xilinx FPGAs
  - Alveo U280: 2x100G port, 8GB HBM2 memory (3.2 Tbps bandwidth), 32GB DDR4, 1.2M logic cells

Xilinx Kintex UltraScale (FPGA)+

Xilinx U280 FPGA card

ESnet

Telemetry Producers



Payload removed

Copy of original packet of a TCP flow

Programmable Data Plane
Transforms packets

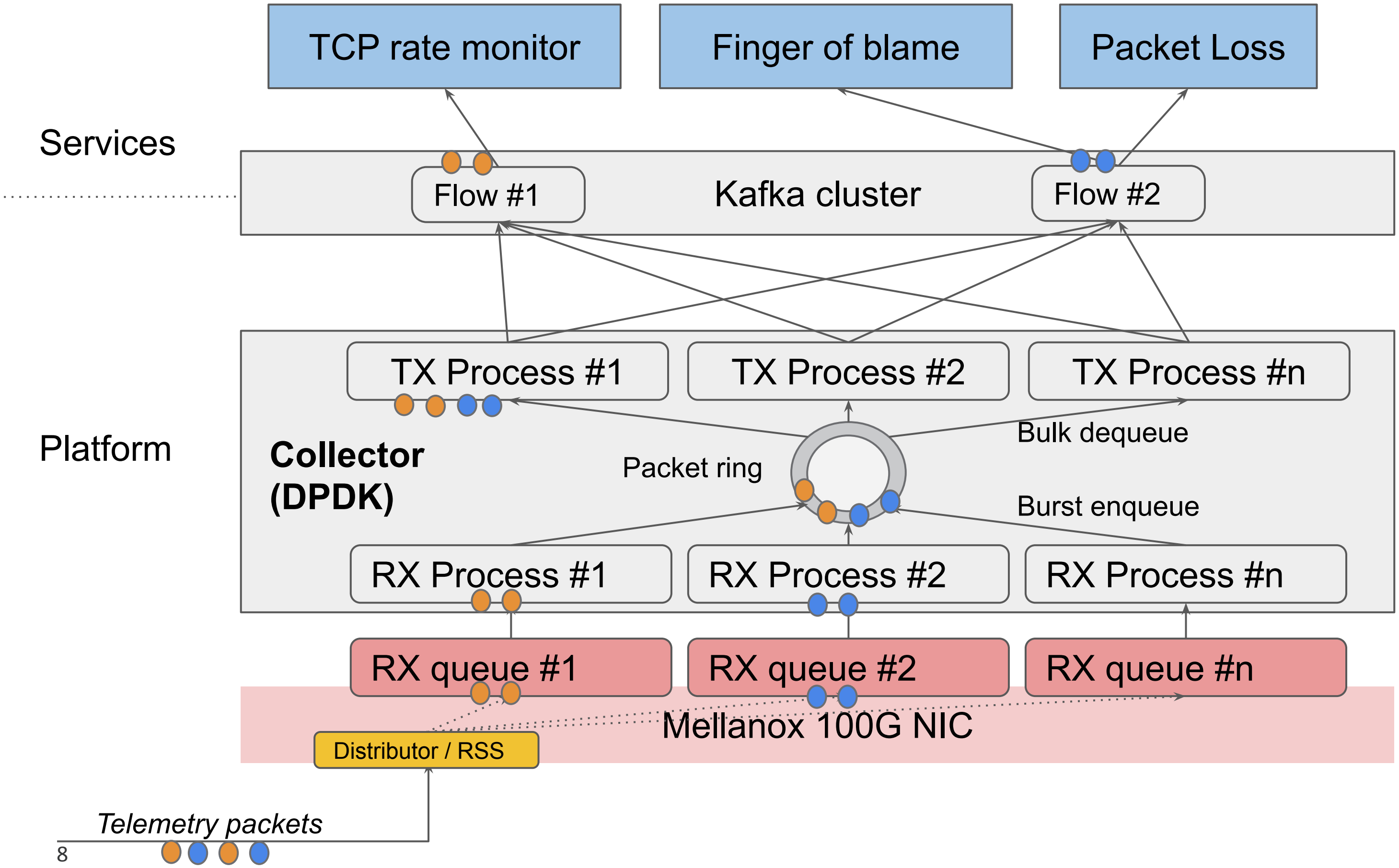
High-Touch Telemetry Packet

Packet size	Rate	Telemetry PPS	Telemetry Rate
1500B	10Gb/s	812K	1,079Mb/s
1500B	100Gb/s	8,127K	10,790Mb/s
9000B	10Gb/s	138K	183Mb/s
9000B	100Gb/s	1,383K	1,833Mb/s

```
type HighTouchLayer struct {
        Version           string
        SensorID          uint8
        VlanId             uint16
        IngressTimestamp  uint64

        // IP header of original packet
        IpSrcAddr        net.IP
        IpDstAddr        net.IP

        // TCP header of original packet

        TcpSrcPort   uint16
        TcpDstPort   uint16
        TcpSeqNo     uint32
        TcpAckNo     uint32

        // Aggregate counters
        FlowPktCount  uint64
        FlowByteCount uint64
        FlowId         uint16
        Flags          uint8
}
```

High-Touch Telemetry Record (approximate) ~100 bytes

High Touch Collector Processing



Topics deleted after 24hr automatically

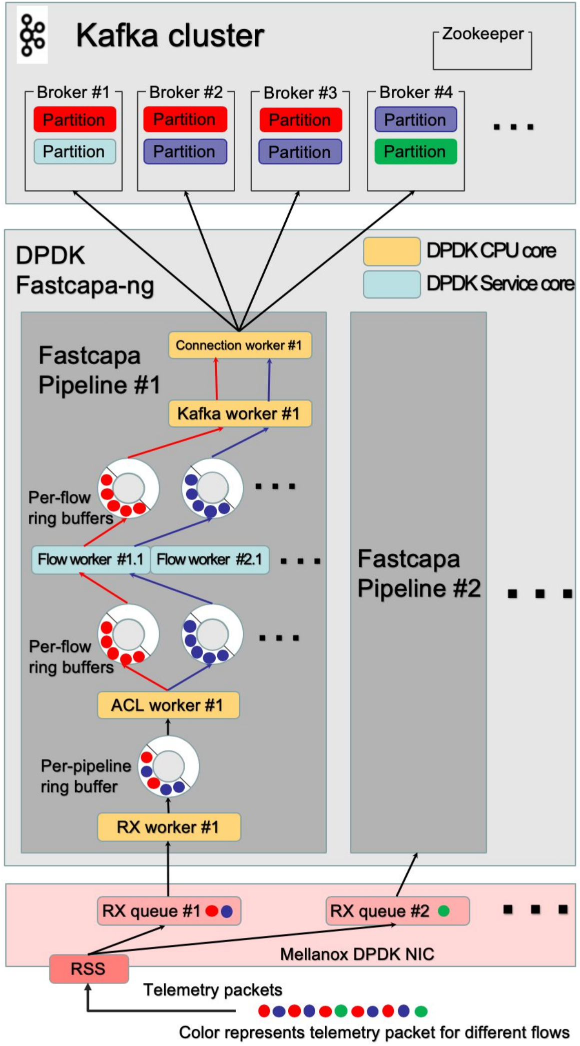
Fastcapa runs on multiple cores (each RX and TX process is one core)

Any DPDK-enabled card

ESnet

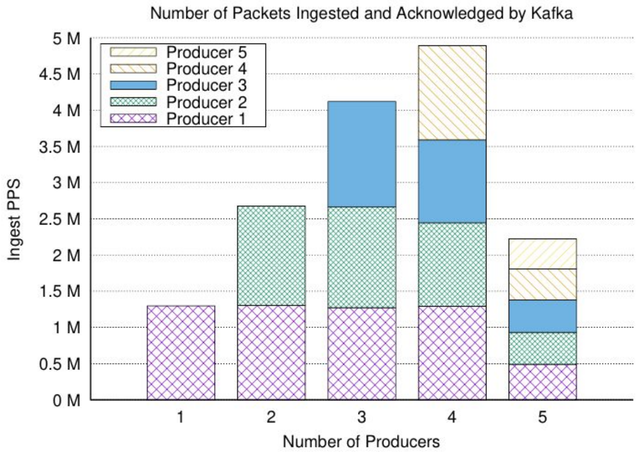
ESnet Fastcapa-ng Internals

- RX queue:
  - NIC dma packets into memory
  - RSS (Receive Side Scaling) applied
- RX worker:
  - pull packet into ring buffers
- ACL worker:
  - classify flows and send them to dedicated rings.
- Flow worker (service cores):
  - process flows using different function: passthrough, sampling, histogram, etc.
  - Flexible N to M mapping of flow to service cores.
- Kafka worker:
  - Combine multiple telemetry packets into large kafka messages.
- Dedicated Kafka connection:
  - maintain TCP connection, message compression task.



Kafka setup and benchmarking

- Docker-compose: bitnami/kafka, JMX Exporter, Prometheus, Grafana
- 6 brokers on a single server
- Possible bottlenecks:
  - Librdkafka C client (inside Fastcapa-ng)
  - Docker proxy - network
  - CPU - Fastcapa and Kafka brokers are on the same host



~5M PPS ingest
untuned single server / 6 broker

Use Case #1
TCP Rate and Retransmission Tracking

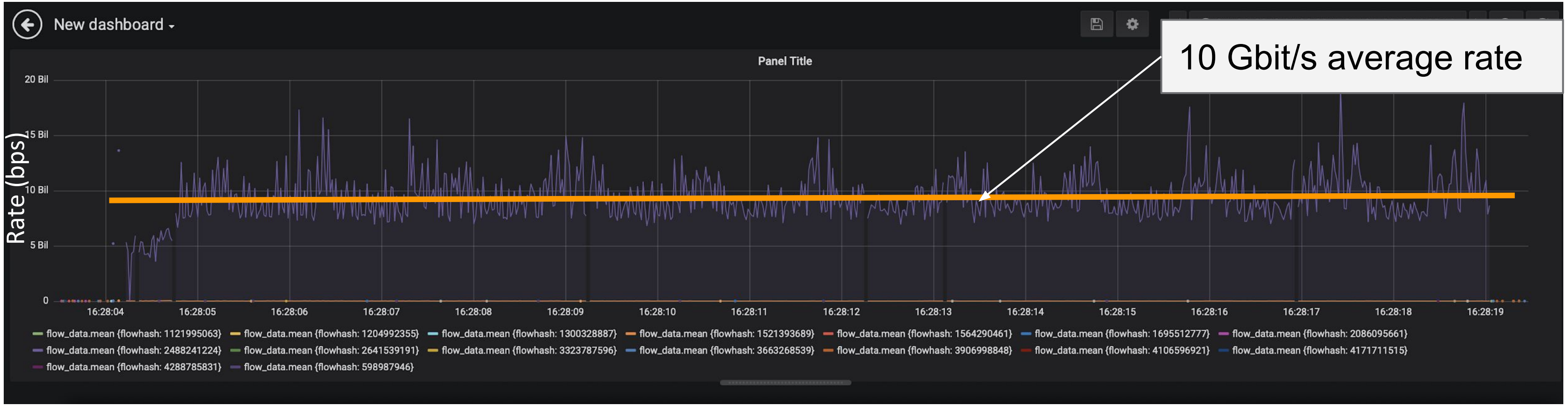
Motivation:

- Monitoring TCP rate in a per-packet basis
    - Find peaks, abnormal rate in the shortest possible time
- Provide a tool for network operations and engineering
- Finding packet retransmissions as they happen
    - Is there an issue at ESnet or at the source or destination networks?

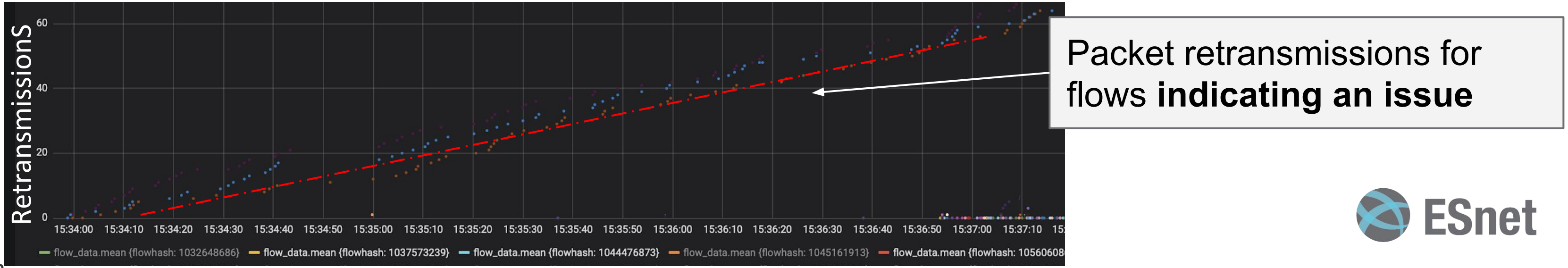
ESnet

Visualizing Real-Time Telemetry Data

- We can plot metrics for every packet in a flow using InfluxDB / Grafana

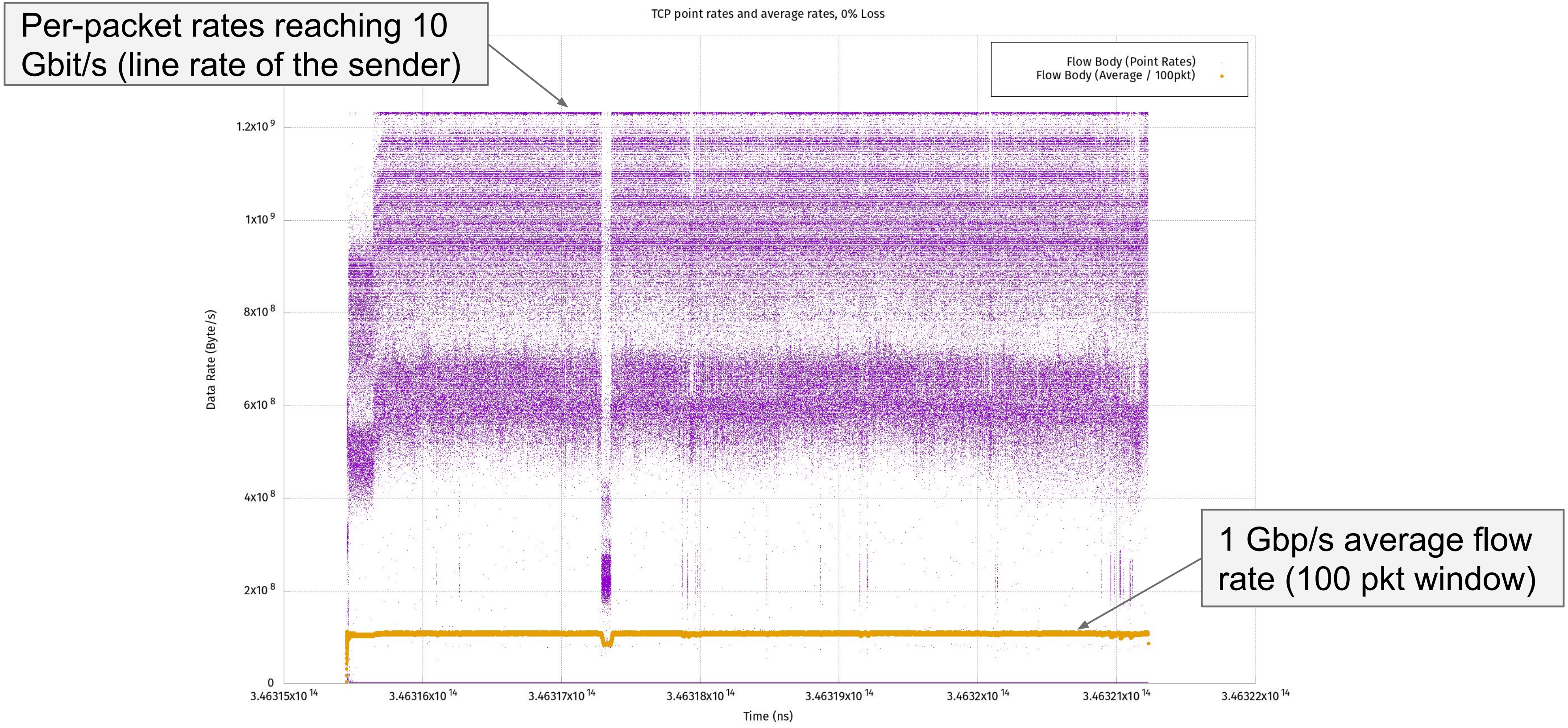


A sample PerfSonar 10Gbit/s test measured by High Touch Rate Monitor



ESnet

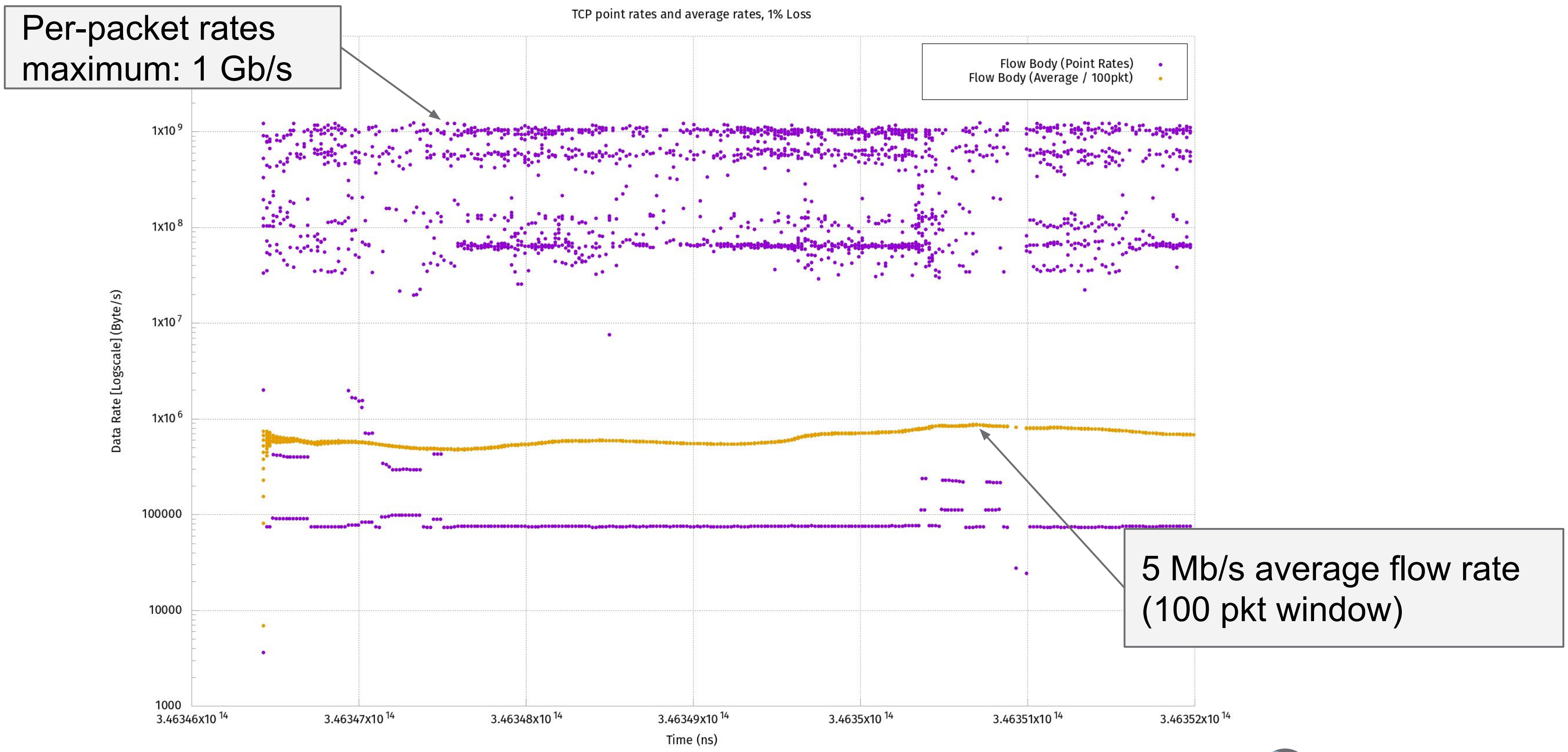
1 Gbps iPerf Flow - 600,000 Packets



Note: Average rate is calculated using a time-weighted average of per-packet rates.

ESnet

1 Gbps iPerf flow - 1% packet drop

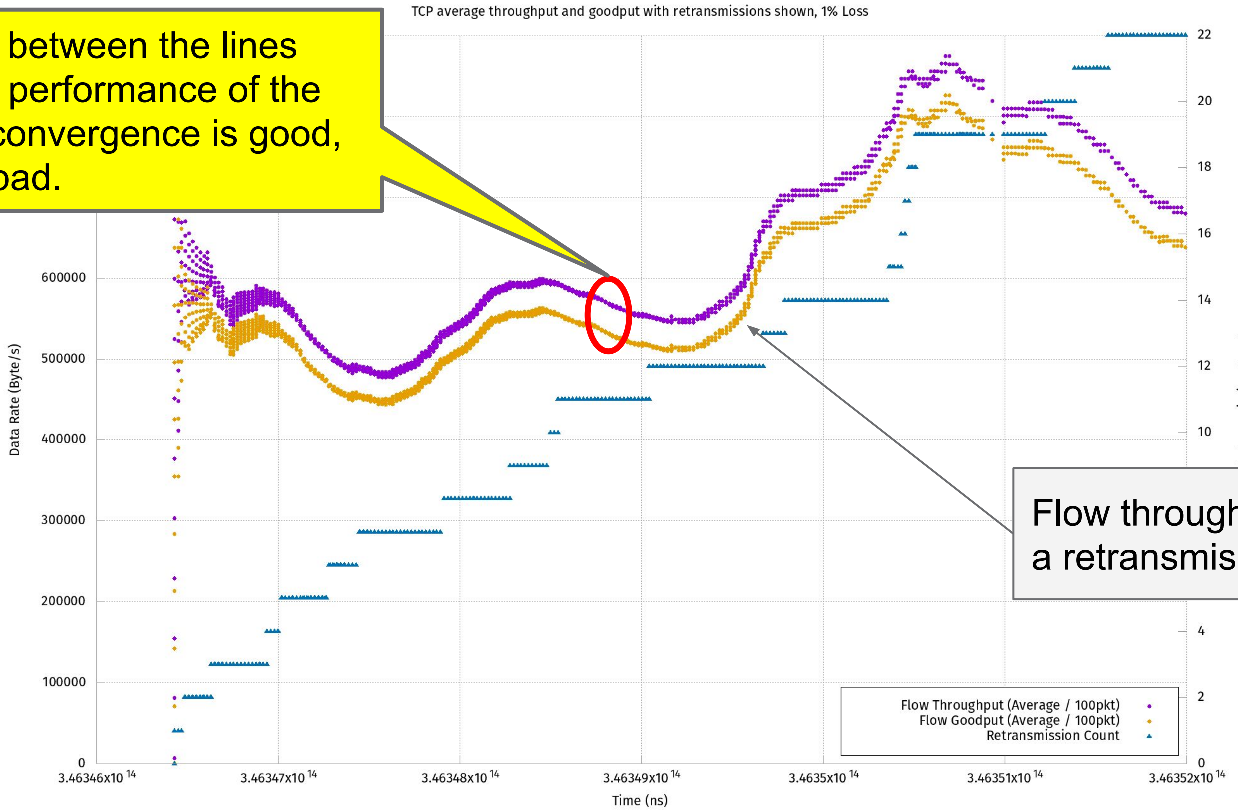


Note: only 23 packets were dropped all together, taking bandwidth down to 5 Mb/s from 1 Gb/s.

ESnet

1 Gbps iPerf Flow - 1% Packet Drop

The difference between the lines represents the performance of the data transfer, convergence is good, divergence is bad.



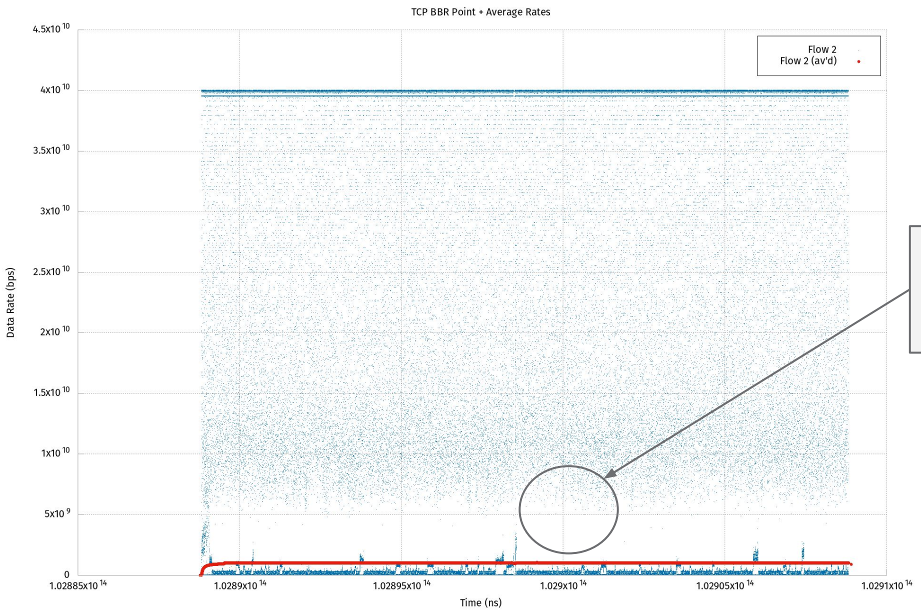
Flow throughput drops when a retransmission happens

Use Case #2
TCP Congestion Control Identification

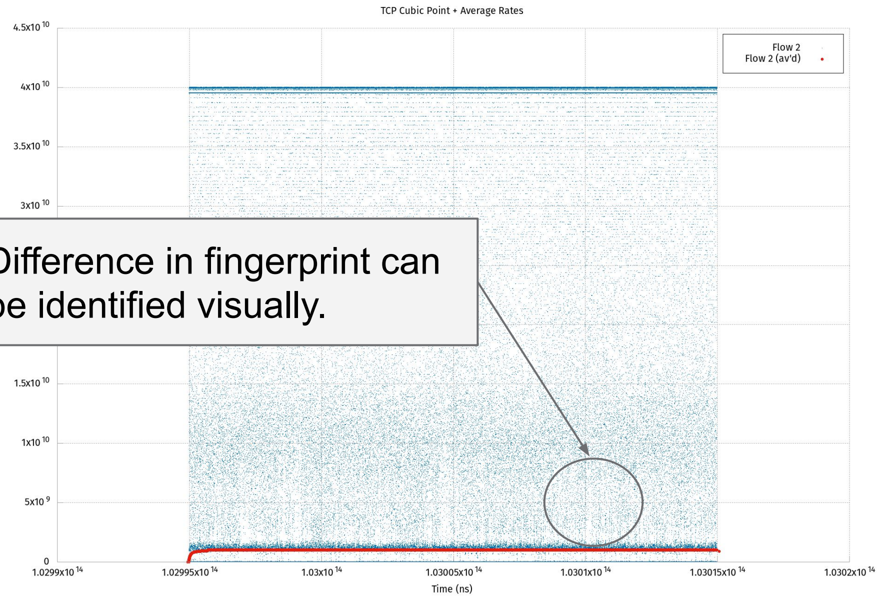
- Motivation:
  - Some flows are unable to utilize the available bandwidth
  - TCP flows can take more of their fair-share
- Discovering misconfigured flows (e.g., window parameters, congestion control) will allow us:
  - Tune the configuration of Data Transfer Nodes
  - Notifying our sites automatically (periodic reports) on suboptimal configuration
  - Guide fair usage of the network ("is equal bandwidth share" fair?)

ESnet

BBR vs Cubic - Point Rates



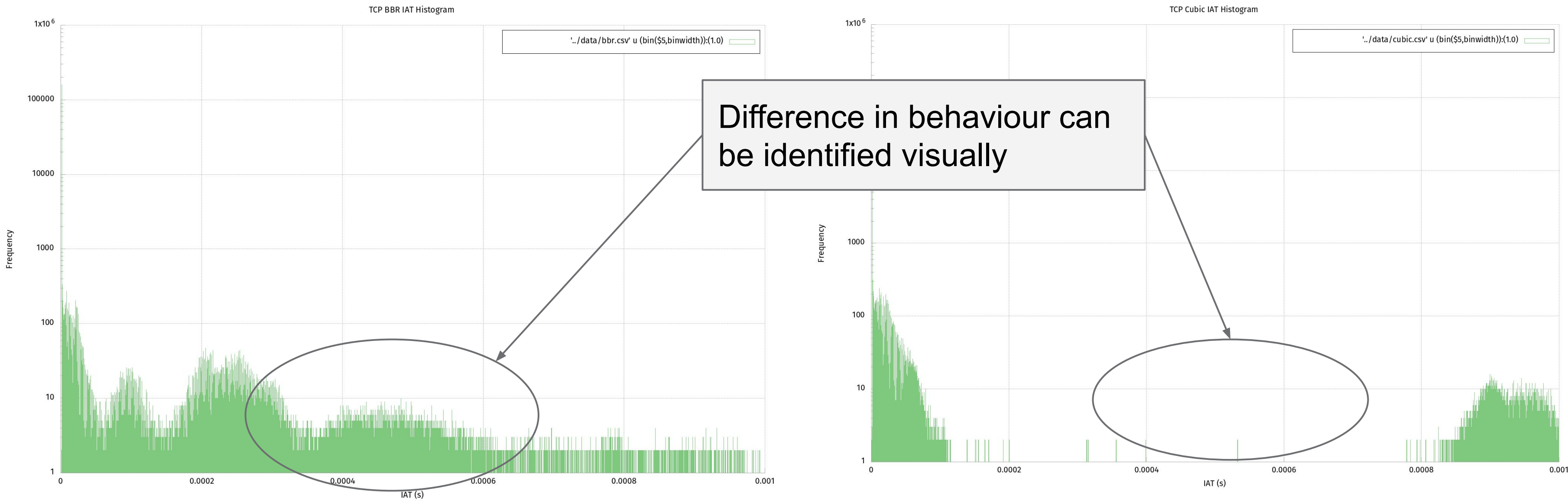
TCP BBR



TCP Cubic

2 millions of data points shown (around 600.000 points a second generated)

BBR vs Cubic - Inter-Arrival Time Histogram



TCP BBR (delay-based)

TCP Cubic (loss-based)

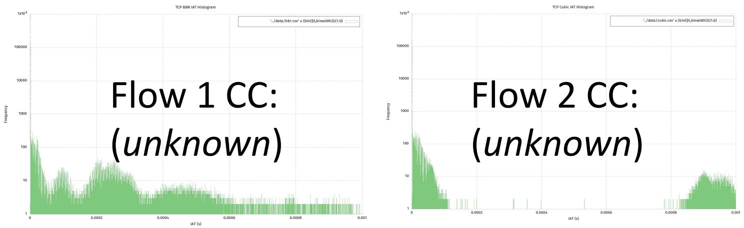
BBR: inter-packet timing is more widespread than other congestion control algorithms.

Machine Learning on Aggregated Data

- Aggregated data - such as histograms can be used to tell apart congestion control (CC) used by TCP flows
- We are using data plane histograms of inter-arrival times per flow (2000 packets per histogram)
- ML algorithms explored: Convolutional Neural Networks, k-Nearest Neighbors

More details, dataplane architecture, ML code in:

Simpson, Kyle A., Richard Cziva, and Dimitrios P. Pezaros. "Seiðr: Dataplane Assisted Flow Classification Using ML." IEEE GLOBECOM, Taipei, Taiwan (2020).



Input: per-flow histograms of Inter-Arrival Time (IAT)

Machine Learning
(trained with labeled data)

Inference in less than 1 ms in all cases

Flow 1 CC: most likely *TCP BBR*

Flow 2 CC: most likely *TCP RENO*

ESnet

High Touch Application Programming

- High Touch Applications can be implemented using **Kafka Streams** - an easy way to program real-time applications on stream of data.
- Expressive, highly scalable and fault tolerant API that allows: aggregation, filtering, counting, grouping data...

APACHE kafka
A distributed streaming platform

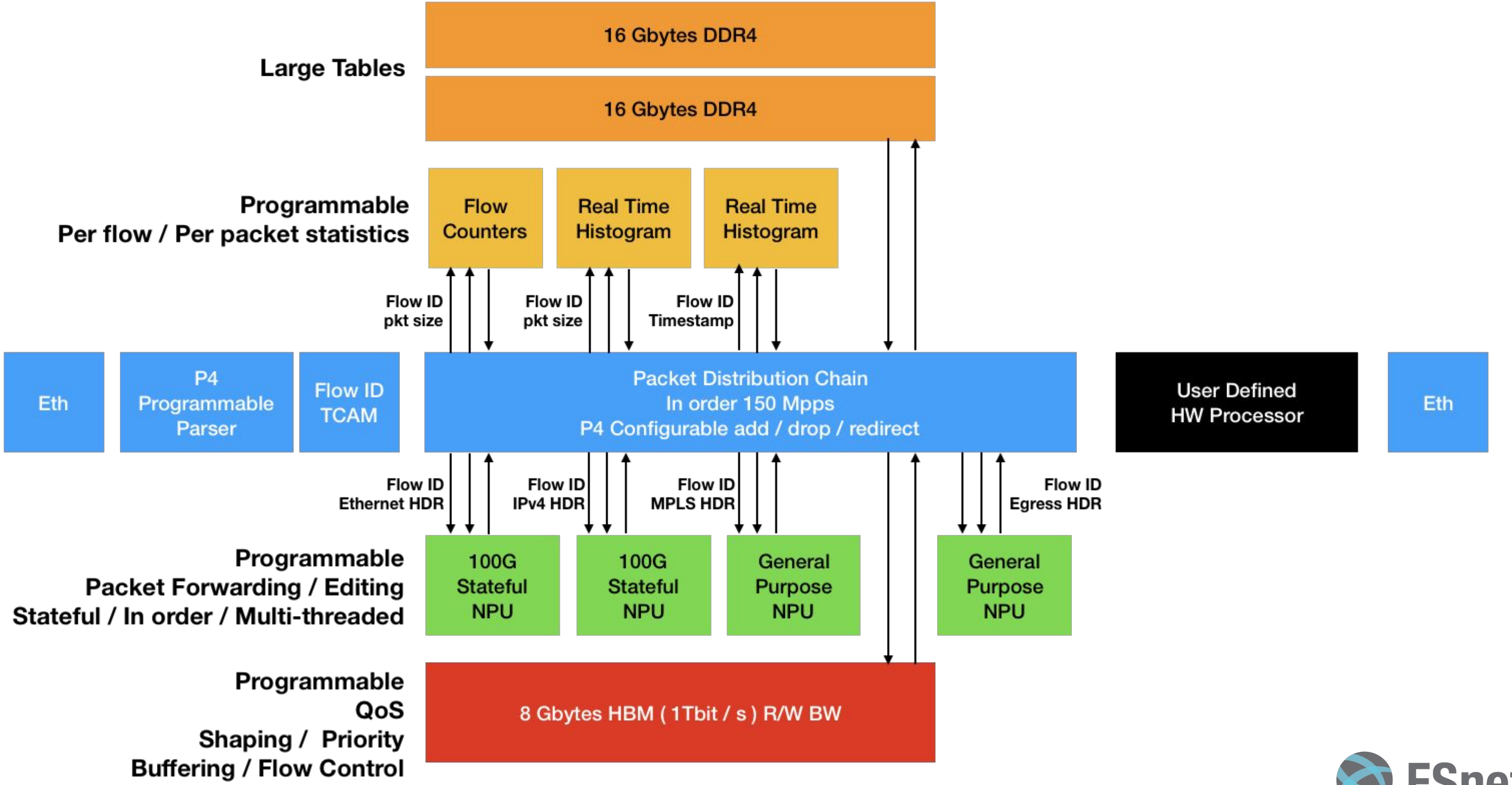
```
int THRES = 10;
KTable<Windowed<String>, Long> SYNcounts = stream
      .filter((k, telemetry) -> telemetry.isSYN())
      .groupBy((k, telemetry) -> telemetry.getIPDstAddr())
      .windowedBy(TimeWindows.of(Duration.ofSeconds(5)))
      .count(Materialized.with(String(), Long()))
      .filter((key, value) -> value > THRES);
SYNcounts.toStream().to("syn-attacks");
```

Example: High Touch SYN Flood Detection

ESnet

High Touch Services
DEMO

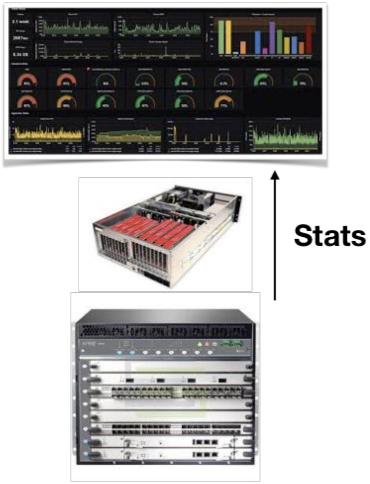
ESnet FPGA Block Diagram - Present and Future



3 models for using FPGAs

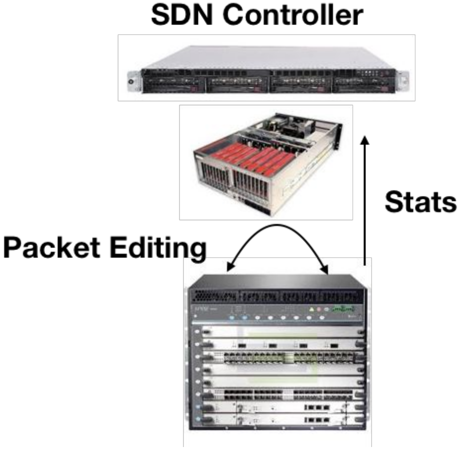
Easy

Install a copy of ESnet's telemetry solution. Zero FPGA development. Customize Splunk / Kentik / Grafana / ELK etc..



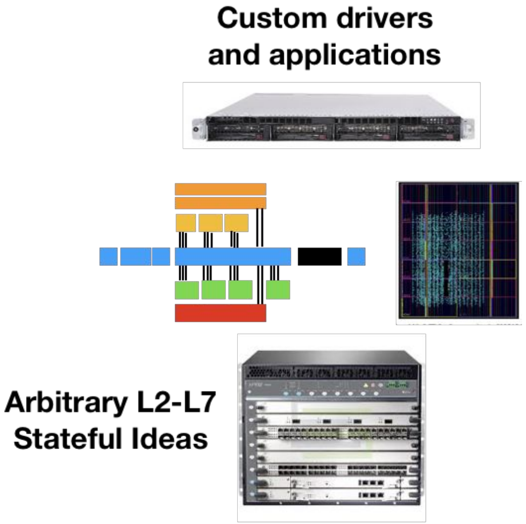
Intermediate

Program the embedded NPUs. Zero FPGA development. FPGA bit file provided. But packet editing is programmable like an SDN switch.



Advanced

Re-configure the FPGA using P4 and Verilog. User defined hardware.



ESnet

High Touch Services Timeline

2019 Q3 — Service Design

Technical service design, experimentation with dataplanes and collector software.

2020 Q1 — Design Validation

Evaluation of the collector software, dataplanes, scoping and prototyping.

2020 Q4 — Design Refinement

Making the service more robust, implementing a variety of High Touch services, while enhancing scalability, fault tolerance, security, orchestration.

2021 Q1 — Pre-Pilot

Deploying pilot service, inspecting traffic on selected links (low-traffic customer, high-traffic customer, Splunk integration).

2022 Q1 — Pre-Deployment

Finalizing a complete solution: edge hosts, programmable hardware, services and their orchestration.

2022 Q3 — Deployment

Deploying High Touch services.

ESnet

Questions…

{bmah, richard, yak}@es.net